Deep Reinforcement Learning-Assisted Energy Harvesting Wireless Networks

Junliang Ye, *Member, IEEE*, and Hamid Gharavi, *Life Fellow, IEEE*

***Abstract*—Heterogeneous ultra-dense networking (HUDN) with energy harvesting technology is a promising approach to deal with the ever-growing traffic that can severely impact the power consumption of small-cell networks. Unfortunately, the amount of harvested energy, which depends on the transmission environment, is highly random and difficult to predict. Since there may be multiple sources of energy in the HUDN, e.g., macro base stations or TV towers, the challenging issue is when and where to harvest energy. Optimally controlling the HUDN can profoundly influence the performance of both data transmission and energy harvesting. However, the working pattern of individual small cell base stations needs to be determined in every time slot. To find an optimal solution in a highly random environment we propose reinforcement learning methods, such as deep deterministic policy gradient (DDPG) and wolpertinger DDPG (W-DDPG). Since the action space is large and discrete for the controlling tasks, a W-DDPG algorithm has been found to be the best approach. The simulation results verify that, compared with the original DDPG algorithm and deep Q-learning, the proposed W-DDPG method can achieve a superior performance in terms of both energy efficiency and throughput.**

***Index Terms*—Reinforcement learning, DDPG, heterogeneous network, energy harvesting, mmWave.**

I. Introduction

HETEROGENEOUS ultra-dense network (HUDN) is emerging as an inevitable solution for fifth and sixth generation (5G & 6G) cellular systems. It enables the transmission of millimeter waves to accommodate growing numbers of users with higher data rates [1]. However, due to the relatively limited range of the millimeter wave, service providers have begun the process of cell densification in existing networks. This demands a significant increase in the number of small cell base stations (SBSs) installation. Although SBSs consume less power compared to regular macro base stations (MBSs), the anticipated massive increase in the number of SBSs within small geographical locations would require an unlimited access to power supplies, which cannot always be available [2]. The challenge is not only the cost and time involved, but also getting a power drop to each individual SBS instead of entirely relying on battery backup in space-constrained urban locations. Energy harvesting (EH) is considered to be a critical technology that can significantly improve the energy efficiency of HUDNs. With energy harvesting technology, the SBSs in the HetNets can obtain energy from radio frequency (RF) signals, store it in batteries, and then use it for data transmission [3]. Since the quality of wireless links and the location of each users equipment (UE) changes from time to time, the main challenge is how to control an EH-assisted HUDN within each time slot in order to improve the performance of energy efficiency. For a time slotted EH architecture, every base station needs to determine its action at each time slot [4]. Under these conditions, controlling EH-assisted networks is difficult to solve by regular optimization methods, such as convex optimization. The reason is that most of these methods are offline since they need to know the precise values of all involved parameters [5]. Bear in mind that accurately predicting these parameters would be essential, but difficult, since EH-assisted devices are randomly distributed. In order to derive practical online energy management algorithms, the Markov Decision Process (MDP) has been widely utilized in EH communications [6]–[11]. While MDP is an effective tool to solve the control problem in an EH-assisted network, it still faces the problem of dimensionality when the number of parameters is large.

With the assistance of artificial intelligence (AI), solving energy harvesting problems has recently entered a new phase. For instance, a deep-learning-based architecture has been proposed in [12] to aid channel estimation in EH-assisted wireless networks. The authors of [13] leverage the deep feedforward neural network to maximize effective secrecy throughput. In addition, two machine learning techniques, linear regression (LR) and decision trees (DT), have been investigated in [14] to model the harvested energy based on spectral power measurements in real-time. To study an optimal transmission policy for energy-harvesting of wireless sensor nodes, a three-layer monotone neural network has been considered in [15]. The authors of [16] apply a deep belief network (DBN) based approach to solve a joint resource allocation problem for the downlinks of a simultaneous wireless information and power transfer (SWIPT) enabled multi-carrier non-orthogonal multiple access (MC-NOMA) system.

Recently, reinforcement learning (RL) technology has attracted worldwide attention. It deals with learning tasks that require an agent to interact with working environments [5], [17]–[30]. Based on the RL technology, agent interactions provide a unique ability to solve many types of EH problems. This is mainly because most of the uncontrollable

Manuscript received August 7, 2020; revised October 19, 2020 and November 23, 2020; accepted December 8, 2020. Date of publication December 15, 2020; date of current version May 20, 2021. The editor coordinating the review of this article was J. Yang. *(Corresponding author: Junliang Ye.)*

The authors are with the National Institute of Standards and Technology, Gaithersburg, MD 20899 USA (e-mail: junliang.ye@nist.gov; hamid.gharavi@nist.gov).

Digital Object Identifier 10.1109/TGCN.2020.3045075

parameters that influence the performance of networks with EH technology require interactions between the agent and the working environment. Q-learning is a value-based RL algorithm and has been used in many studies [17]–[20]. For example, to guarantee network energy efficiency while ensuring low packet loss probability, the authors in [18] successfully apply a Q-learning to multi-hop deflection routing for EH of nanonetworks. A fuzzy Q-learning algorithm to handle the power management of EH-assisted wireless sensors by interacting with the environment has been studied in [19]. To satisfy quality of service (QoS) constraints over multi-hop relay networks, a Q-learning based optimal routing and power allocation method has also been investigated in [20].

An advanced version of Q-learning is deep Q-learning (DQL), which adopts deep neural networks to evaluate state value functions. This characteristic enables a DQL-based approach to solve many complicated tasks [21]. In addition, [22] also applies a DQL algorithm to support an EH-assisted network by interacting with the environment in order to maximize utility within the uncertainties of harvested energy, request arrivals, and resource prices. To optimize the energy efficiency while maintaining QoS, [23] proposes a DQL-based framework for dynamic resource allocation in EH-assisted networks. A deep distributed recurrent Q-network algorithm is proposed to manage the complex dynamic channel, data, and energy environment through a partially observable state [24]. To maximize network throughput performance, [25] investigates a DQL-based optimal policy for transmission power allocation. In this method the modulation level is adjusted adaptively according to the obtained causal information on harvested energy, battery state, and channel gain. To simultaneously maximize the throughput and minimize the prediction inaccuracy of the battery level of EH devices, a two-layer RL network is adopted in [26].

It should be noted that in order to successfully apply RL methods, such as Q-learning and deep Q-learning to solve stochastic optimization problems, it is essential to discretize all continuous variables of the state and action scenarios into a finite set of discrete values [27]. These requirements however, limit their applications as most of the involved parameters have continuous values. To overcome this, an RL-based algorithm called deep deterministic policy gradient (DDPG) is proposed in [28]. In this article, the authors present an actor-critic, model-free algorithm based on the deterministic policy gradient that can operate over continuous state and action spaces. Based on DDPG, the authors of [29] propose a joint optimization scheme for data transmission delay, energy consumption, and bandwidth allocation in an EH-assisted network. To optimally control the power, a DDPG algorithm, without prior knowledge of energy arrival, user arrival, and channel state information, has been studied in [30]. In addition, the authors of [5] propose a DDPG-based algorithm applicable for continuous states suitable for continuous energy management.

As a newly developed RL technology, there are some studies using DDPG to solve tasks with continuous action and state spaces. However, the control problem of base stations in an EH-assisted network is a task with continuous state space, but discrete action space. Moreover, as the number of SBSs in the network increases, the action space rises dramatically. Therefore, in this article we propose a Wolpertinger architecture-based deep deterministic policy gradient (W-DDPG) to maximize energy efficiency in a EH-assisted HUDN. W-DDPG can avoid the problem of dimensionality compared with Q-learning, which requires discretization of the state. On the other hand, W-DDPG can solve the nonconvex objective function in a long-term average form, which is crucial to improving the lifetime of SBS's. The major contributions of this article are as follows.

1) We formulate the energy efficiency optimization problem in an EH-assisted HUDN as an RL problem to maximize long-term average energy efficiency and this basically requires defining the state, action, and reward of the RL framework. Thus, to solve the optimization problem we map the parameters that influence the performance of the HUDN into state, action, and reward forms.
2) The W-DDPG based framework is developed to achieve an optimal learning policy with continuous state and large-scale discrete actions. As the DDPG algorithm can only be used to perform continuous actions, it's impossible to apply it directly. Also, a simple discretization method like the floor or round function, is not suitable for discretizing the actions in DDPG. Therefore, we adopt a k-nearest-neighbor (k-NN) algorithm-based method to perform discretization on the DDPG actions in order to improve the performance of the HUDN. To the best of our knowledge, this is the first time that a W-DDPG-based algorithm has been considered as a control for EH-assisted HUDN.
3) Simulations have been carried out to evaluate the performance of the proposed W-DDPG architecture. Since convergence of the deep RL algorithm can be strongly impacted by configuration of hyperparameters, a series of parameters have been evaluated to further improve the performance of the algorithm. Compared with the DQL algorithm and the simple action discretization method assisted DDPG algorithm, the simulation results verify that our proposed W-DDPG-based algorithm can effectively improve the energy harvesting and throughput performance of EH-assisted HUDNs.

The rest of the article is organized as follows: The system model it described in detail in Section II, including definitions of working patterns of base stations, analytical models of power consumption, energy harvesting and data transmission. Section III introduces the W-DDPG RL architecture that is used in this article. Simulations of the proposed algorithm are carried out in Section IV. Conclusions are finally drawn in Section V.

II. System Model

A. Network Architecture

Here, we consider a EH-assisted HUDN where only MBSs are connected to the power grid. In other words, all SBSs in

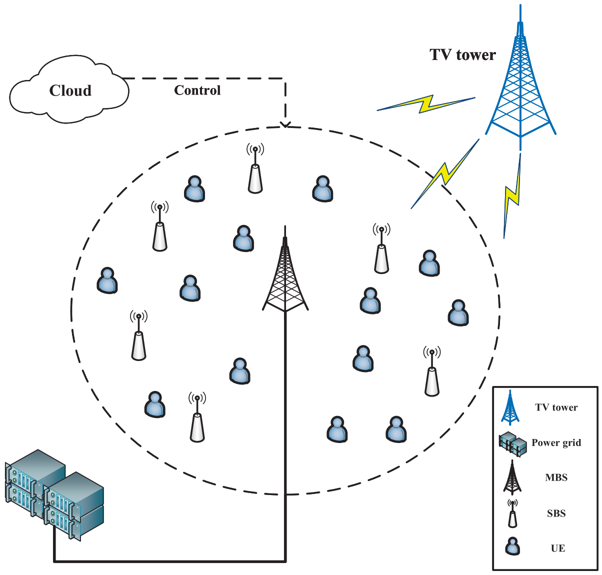

Fig. 1. Network architecture.

this network need to harvest energy to serve the user equipment (UE). Also, MBSs can transmit energy to SBSs through wireless links where the transmitted energy will be harvested by SBSs and stored in their batteries. In addition, each SBS can harvest energy from TV white signals, which are broadcasted by TV towers. An agent at the cloud server controls operations of the MBSs and SBSs, as shown in Fig. 1.

Let's define ϕ_{MBS}, ϕ_{SBS}, ϕ_{UE} and ϕ_{TV} as the set of MBSs, SBSs, UEs, and TV towers, respectively. Then, the number of elements contained in these sets is defined as $\lfloor \phi_{MBS} \rfloor$, $\lfloor \phi_{SBS} \rfloor$, $\lfloor \phi_{UE} \rfloor$ and $\lfloor \phi_{TV} \rfloor$ respectively, where $\lfloor \cdot \rfloor$ represents the cardinality of the corresponding set. Thus, these four sets are represented as $\phi_{MBS} = \{mb_j | 1 \leq j \leq \lfloor \phi_{MBS} \rfloor\}$, $\phi_{SBS} = \{sb_k | 1 \leq k \leq \lfloor \phi_{SBS} \rfloor\}$, $\phi_{UE} = \{ue_i | 1 \leq i \leq \lfloor \phi_{UE} \rfloor\}$, and $\phi_{TV} = \{tv_n | 1 \leq n \leq \lfloor \phi_{TV} \rfloor\}$. The MBSs and SBSs are assumed to be uniformly distributed in a two-dimensional plane $\mathbb{A}_T$. The TV towers are located outside $\mathbb{A}_T$. The UEs' movements are assumed to follow a random walk mobility model as follows,

$$\begin{cases} xu_i\left(t_g + \Delta t_s\right) = xu_i\left(t_g\right) + \upsilon_{xi}\left(t_g\right) \\ yu_i\left(t_g + \Delta t_s\right) = yu_i\left(t_g\right) + \upsilon_{yi}\left(t_g\right) \end{cases}, \tag{1}$$

where $xu_i(t_g)$ and $yu_i(t_g)$ are the coordinates of the location of UE: ue_i, at a given time slot t_g. Both $\upsilon_{xi}(t_g)$ and $\upsilon_{yi}(t_g)$ in (1) are two normally distributed random variables with expectation and variance as 0 and 1, respectively.

B. Working Patterns of Base Stations

In this model, if an SBS: sb_k, is the closest base station to a UE: ue_i, we then assume it is covered by sb_k. Also, if an MBS: mb_j, is the closest base station to ue_i, or sb_k, the case is the same. We should point out that if a UE is covered by an SBS, it doesn't mean that it will be associated with this SBS because of the status of battery or the action that is taken by the corresponding SBS, i.e., if a UE is covered by an SBS, which takes an action to harvest energy, then the UE will not be associated with this SBS. Notice that we consider a widely used distance-based association strategy in order to reduce the complexity of problem formulation. The system is considered to operate in a time-slotted fashion with an equal-length time slot represented as: Δt_s. At the beginning of each time slot, SBSs can switch between a data transmission mode and energy harvesting mode according to conditions like battery level and the amount of communication requests from UEs. When an SBS is in energy harvesting mode, there are two sources of energy that they can rely on: MBSs and broadcasting signals from TV towers.

According to [6], [31], the charging speed of analog beamforming transmission is faster when compared with omnidirectional transmission. However, it's more energy consuming compared with omnidirectional transmission. On the other hand, utilizing TV white signals is a potential solution for energy harvesting. This is because TV towers continuously broadcast signals so SBSs can always harvest energy without having to rely on MBSs. Also, the charging speed of the TV signal is higher compared with omnidirectional transmission [6]. Furthermore, harvesting energy from TV towers instead of MBSs can save more energy, since the operation of the TV tower is independent of the HUDN. However, harvesting energy from TV towers depends mainly on transmission environments (e.g., weather conditions, the distance between TV towers and SBSs, etc.) [32]. Therefore, it is difficult for an SBS to determine the best way to harvest the energy.

To address this issue, we assume that at the beginning of each time slot: t_g, one of the following three types of actions can be selected by an MBS: mb_j, and an SBS sb_k. These actions are summarized in Table I.

In this article, the MBSs have two main functions in the network; data transmission and charging SBSs. Our aim is to improve the energy efficiency of HUDNs on the premise of satisfying user communication's needs. Under these conditions, three actions are designed to deal with the following cases.

Case 1: The number of communication requests from UEs is significant, while the average battery level of SBSs is high. In this case, there are only a few charging requests from SBSs, despite many communication requests from UEs. The M1 action is designed to deal with this case.

Case 2: The number of communication requests from UEs is small, while the average battery level of SBSs is high. When an MBS is transmitting signals with an omnidirectional antenna, all the associated SBSs with the S2 action can receive energy from the transmission. Moreover, although the data rate will be lower compared with M1 action, better energy saving can be achieved without hybrid precoding.

Case 3: The number of communication requests from the UEs is large, while the battery levels of some SBSs are low. We consider directional transmission and reception with analog processing and phase shifters in the M3 action. According to [31], it's much more energy efficient to charge SBSs with analog beamforming than hybrid beamforming as the power consumption of RF chains is substantially higher. Compared with omnidirectional transmission, analog processing based directional transmission can quickly charge the associated

TABLE I
ACTIONS OF MBSS AND SBSS

M1 action	mb_j will transmit signals on the mmWave band to associated UEs with hybrid beamforming and massive MIMO antennas
M2 action	mb_j will transmit signals on the sub-6GHz spectrum to associated UEs with an omnidirectional antenna
M3 action	mb_j will transmit signals on the mmWave band to associated UEs with hybrid beamforming and massive MIMO antennas while charging SBSs with S2 action by analog beamforming
S1 action	sb_k will harvest energy from TV signals
S2 action	sb_k will harvest energy from the MBSs
S3 action	sb_k will transmit signals on the mmWave band to the associated UEs with hybrid beamforming and massive MIMO antennas

SBSs. However, as one-directional beam can only charge one SBS, action M3 consumes more energy compared with M2 action, especially when the number of charging requests from SBSs is large.

Case 4: The number of communication requests from UEs is small, while the battery levels of some of the SBSs are low. In general, this case is caused by a large number of users associated with only a few SBSs. MBSs, which are closest to low battery SBSs, can select M3 action for higher charging speed, whereas the low battery SBSs proceed with the S2 action.

It's obvious that any action taken by an MBS and the transmission conditions of TV signals can strongly influence the amount of harvested energy of sb_k. Therefore, it is still difficult for an SBS to decide which action to take. To solve this problem, we propose an RL-based scheme, which will be described in Section III.

C. Communication Requests and Base Station Associations

In this article, the association strategies for UEs are configured as follows,

1) The UE with a communication request will be associated with the nearest base station.
2) If the UE is associated with an SBS, and the battery level of the SBS is below a given threshold B_{tr}, or the nearest SBS is in the energy harvesting pattern, the UE will be associated with the nearest MBS of this SBS.

We assume that at the beginning of each time slot, each UE will have a decision with a probability, P_{cr}, about whether to initiate a communication request. If a UE: ue_i, is successfully associated with an SBS, the duration of the association dr_i is assumed to follow an exponential distribution with an expectation of 1 [33]. After this duration, ue_i will finish the association. Assuming communication requests from UEs to be independent, for a given SBS sb_k, the number of communication requests at a given time slot: t_g, can be expressed as,

$$Ns_k(t_g) = \left(\left(\sum_{i=1}^{nu_k(t_g)} Cr_i(t_g)\right)\mathbf{1}(bl_k(t_g) \geq B_{tr}) + Ns_k(t_g - \Delta t_s) - Ls_k(t_g)\right) \times \mathbf{1}(as_k(t_g) = \mathrm{S3}), \tag{2}$$

where

$$Cr_i(t_g) = \begin{cases} 1 & ue_i \,\text{has a communication request} \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

$bl_k(t_g)$ is the battery level of sb_k at time slot t_g, $\mathbf{1}(\cdot)$ is the indicator function, and $nu_k(t_g)$ represents the number of UEs that are covered by sb_k at t_g. $Ls_k(t_g)$ is the number of UEs that finish the association at t_g, and $as_k(t_g)$ is the action that is taken by sb_k at t_g. Thus, for a given MBS: mb_j, the number of communication and charging requests at a given time slot t_g can be expressed as (4), shown at the bottom of the page, where $ns_k(t_g)$ is the number of SBSs that are covered by mb_j, $nm_j(t_g)$, $nm_j(t_g)$ is the number of UEs that are covered by mb_j, and $Lm_j(t_g)$ is the number of UEs that finish the association at t_g.

D. Power Consumption

We assume that the power consumption of base stations consists of two parts: operational power consumption (e.g.,

$$\begin{aligned} Nm_j(t_g) = &\left(\sum_{l=1}^{ns_k(t_g)}\left(\sum_{i=1}^{nu_k(t_g)} Cs_i(t_g)\right)\mathbf{1}(bl_k(t_g) < B_{tr}) \right.\\ &\left. + \sum_{i=1}^{nm_i(t_g)} Cr_i(t_g) + Nm_j(t_g - \Delta t) - Lm_j(t_g)\right)\mathbf{1}(am_j(t_g) = \mathrm{M1 \ or \ M2}) \\ &+ \left(\sum_{l=1}^{ns_k(t_g)}\left(\sum_{i=1}^{nu_k(t_g)} Cs_i(t_g)\right)\mathbf{1}(bl_k(t_g) < B_{tr}) + \sum_{l=1}^{ns_k(t_g)} \mathbf{1}(as_k(t_g) = \mathrm{S2}) \right.\\ &\left. + \sum_{i=1}^{nm_i(t_g)} Cr_i(t_g) + Nm_j(t_g - \Delta t) - Lm_j(t_g)\right)\mathbf{1}(am_j(t_g) = \mathrm{M3}) \end{aligned} \tag{4}$$

the energy consumption baseband, power amplifier, and so on) and constant power consumption (i.e., the energy consumption when there is no traffic load). Based on [34], the power consumption of an SBS: sb_k, during time period $[t_g,\ t_g+\Delta t_s]$ can be represented as,

$$ps_k(t_g) = Ns_k(t_g)\frac{\frac{p_{str}}{\eta_{pa}} + p_{srf} + p_{sbb}}{(1-\sigma_{dc})(1-\sigma_{ms})} \times \mathbf{1}\big(as_k(t_g) = \mathrm{S3}\big) + p_{sst}, \tag{5}$$

where p_{str} is the transmit power consumption of sb_k, η_{pa} is the efficiency coefficient of the power amplify module, p_{srf}, is the power consumption of the radio frequency module, p_{sbb}, is the power consumption of the baseband module, σ_{dc}, is the loss coefficient of the digital control module, σ_{ms}, is the power supply loss coefficient, and p_{sst}, is the constant power consumption, which is independent from the traffic load of sb_k. $Ns_k(t_g)$ is the number of communication requests at t_g. Based on the configuration in [34], the values of σ_{ms} and σ_{dc} are smaller than 1.

Also, the power consumption of an MBS, mb_j, during time period $[t_g,\ t_g+\Delta t_s]$ can be expressed as (6), shown at the bottom of the page, where p_{mhtr} is the transmit power consumption of a hybrid beamforming transmission pattern, p_{motr} is the transmit power consumption of the omnidirectional transmission pattern, p_{mhrf} is the power consumption of the radio frequency module of hybrid beamforming transmission pattern, p_{morf} is the power consumption of the radio frequency module of omnidirectional transmission pattern, p_{mhbb} is the power consumption of the baseband module of hybrid beamforming transmission pattern, p_{mobb} is the power consumption of the baseband module of omnidirectional transmission pattern, p_{mst} is the constant power consumption, which is independent from the traffic load of mb_j, and $Nm_j(t_g)$ is the number of communication and charging requests at t_g.

E. Energy Harvesting

An SBS harvests energy either from MBSs or from TV towers. Based on [32], for a given SBS, sb_k, the amount of energy that can be harvested during the time period $[t_g,\ t_g+\Delta t_s]$ from the TV towers can be expressed as,

$$Et_k(t_g) = \left(\sum_{n=1}^{\lfloor\phi_{TV}\rfloor} p_{tvtr}\cdot 10^{\frac{-Pl_{kn}}{10}}\right) \times \mathbf{1}\big(as_k(t_g) = \mathrm{S1}\big)\Delta t_s, \tag{7}$$

where p_{tvtr} is the transmitting power of the TV towers and Pl_{kn} is the path loss of the transmission link, which is expressed as,

$$\begin{aligned} Pl_{kn} = (&69.55 + 26.16\log ft_n \\ &- 13.82\log Ht_n - a(Hs_k) \\ &+ (44.9 - 6.55\log Ht_n))\log d_{kn}. \end{aligned} \tag{8}$$

where ft_n is the transmission frequency of tv_n, Ht_n is the height of the TV tower tv_n, d_{kn} is the distance between tv_n and sb_k, $a(Hs_k)$ is the correction factor for the height of the receiving antenna, and Hs_k is the height of the receiving antenna of sb_k. For a medium-sized city, $a(Hs_k)$ is given by,

$$\begin{aligned} a(Hs_k) = &(1.1\log ft_n - 0.7)Hs_k \\ &- (1.56\log ft_n - 0.8). \end{aligned} \tag{9}$$

Based on [31], the energy that sb_k harvests from MBSs with action M2 during a given time period $[t_g,\ t_g+\Delta t_s]$ can be expressed as (10), shown at the bottom of the page, where G_{mo} is the antenna gain for omnidirectional transmission, h_{kj} is the small scale fading which follows an exponential distribution with expectation as 1, d_{kj} is the distance between sb_k and an MBS mb_j, and α_p is the path loss exponent.

The authors of [31] indicate that in mmWave networks, interference has little impact on the harvested power. So, we can ignore the effect of interference on harvested energy from the mmWave spectrum. The energy harvested from the MBSs with action M3 during a given time period, $[t_g,\ t_g+\Delta t_s]$, can be expressed as below, where G_{mh} is the antenna gain for hybrid precoding transmissions.

Thus, the energy that sb_k harvests during a given time period $[t_g,\ t_g+\Delta t_s]$ can be rewritten in closed-form as (12),

$$\begin{aligned} pm_j(t_g) = {}& Nm_j(t_g)\frac{\frac{p_{mhtr}}{\eta_{pa}} + p_{mhrf} + p_{mhbb}}{(1-\sigma_{dc})(1-\sigma_{ms})}\mathbf{1}\big(am_j(t_g) = \mathrm{M1}\big) \\ &+ Nm_j(t_g)\frac{\frac{p_{motr}}{\eta_{pa}} + p_{morf} + p_{mobb}}{(1-\sigma_{dc})(1-\sigma_{ms})}\mathbf{1}\big(am_j(t_g) = \mathrm{M2}\big) \\ &+ Nm_j(t_g)\frac{\frac{p_{mhtr}}{\eta_{pa}} + p_{mhrf} + p_{mhbb}}{(1-\sigma_{dc})(1-\sigma_{ms})}\mathbf{1}\big(am_j(t_g) = \mathrm{M3}\big) + p_{mst} \end{aligned} \tag{6}$$

$$Eo_k(t_g) = \left(\sum_{j=1}^{\lfloor\phi_{MBS}\rfloor}\sum_{k=1}^{Nm_j(t_g)}\Big(p_{motr}G_{mo}h_{kj}d_{kj}^{-\alpha_p}\Big)\mathbf{1}\big(am_j(t_g) = \mathrm{M2}\big)\right)\mathbf{1}\big(as_k(t_g) = \mathrm{S2}\big)\Delta t_s \tag{10}$$

$$Eh_k(t_g) = \left(\sum_{j=1}^{\lfloor\phi_{MBS}\rfloor}\Big(p_{mhtr}G_{mh}h_{kj}d_{kj}^{-\alpha_p}\Big)\mathbf{1}\big(am_j(t_g) = \mathrm{M3}\big)\right)\mathbf{1}\big(as_k(t_g) = \mathrm{S2}\big)\Delta t_s \tag{11}$$

shown at the bottom of the page, and the battery level that sb_k can harvest at $t_g + \Delta t_s$ is,

$$bl_k(t_g + \Delta t_s) = bl_k(t_g) + E_k(t_g) - ps_k(t_g)\Delta t_s. \quad (13)$$

Notice that we don't consider the charging and discharging losses when SBSs harvest energy. However, in practical situations the RL agent can directly assess the value of the reward that includes the charging and discharging losses from the environment.

F. Data Transmission

According to the actions of MBSs and SBSs described in Section II-B, there are two downlink data transmission methods in the HUDN: data transmission on the mmWave spectrum with hybrid precoding and massive MIMO antennas, and the other is to transmit data on the sub-6GHz spectrum with the omnidirectional antenna. Thus, at the beginning of a given time slot t_g, a UE, ue_i, may be in one of these four types of association patterns.

Thus, by assuming the throughput of the downlink between typical ue_i and the associated base station equals the channel capacity of the corresponding link, without loss of generality, the throughput of patterns U2 and U4 can be expressed as,

$$\begin{aligned} Tr_{ji,h}(t_g) &= Ba_{mh}\Delta t_s \\ &\quad \times \log_2\left(1 + \frac{p_{mhtr} G_{mh} h_{ji} d_{ji}^{-\alpha_p}}{p_{no}}\right), \quad (14) \\ Tr_{ki}(t_g) &= Ba_{sh}\Delta t_s \\ &\quad \times \log_2\left(1 + \frac{p_{str} G_{sh} h_{ki} d_{ki}^{-\alpha_p}}{p_{no}}\right), \quad (15) \end{aligned}$$

where $Tr_{ji}(t_g)$ is the throughput of the link between ue_i and mb_j during the same time period, $Tr_{ki}(t_g)$ is the throughput of the link between ue_i and sb_k during the time period $[t_g,\ t_g + \Delta t_s]$, Ba_{mh}, Ba_{sh} and G_{mh}, G_{sh}, are the bandwidths, and the antenna gains of the corresponding working patterns of the base stations, respectively. h_{ji} is the small scale fading of the link between ue_i and mb_j. Similarly, h_{ki} is the small scale fading of the link between ue_i and sb_k. Both h_{ji} and h_{ki} are exponentially distributed random variables with expectation as in 1. d_{ji} is the distance between ue_i and mb_j, d_{ki} is the distance between ue_i and sb_k. α_p is the path loss exponent. The noise term, p_{no}, is assumed to be a normally distributed variable with zero expectation and variance σ_n^2. Notice that the interference has not been taken into consideration here because the beam is narrow enough to ignore interference when the data is transmitted on the mmWave spectrum with massive MIMO antennas and hybrid precoding.

However, interference cannot be ignored when the data is transmitted on the sub-6GHz spectrum with the omnidirectional antenna. Thus, the throughput of state U3 can be expressed as (16), shown at the bottom of the page, where h_{zi} is the small scale fading of the link between ue_i and the interfering MBS mb_z. Similarly, d_{zi} is the distance between ue_i and mb_z. Therefore, the throughput of the HUDN during time period, $[t_g,\ t_g + \Delta t_s]$, is expressed as

$$\begin{aligned} Tr_{hn}(t_g) &= \sum_{j=1}^{\lfloor \phi_{MBS} \rfloor} \sum_{i=1}^{Nm_j(t_g)} Tr_{ji,h}(t_g)\mathbf{1}(am_j(t_g) = \text{M1}) \\ &\quad + \sum_{j=1}^{\lfloor \phi_{MBS} \rfloor} \sum_{i=1}^{Nm_j(t_g)} Tr_{ji,o}(t_g)\mathbf{1}(am_j(t_g) = \text{M2}) \\ &\quad + \sum_{j=1}^{\lfloor \phi_{MBS} \rfloor} \sum_{i=1}^{Nmu_j(t_g)} Tr_{ji,h}(t_g)\mathbf{1}(am_j(t_g) = \text{M3}) \\ &\quad + \sum_{k=1}^{\lfloor \phi_{SBS} \rfloor} \sum_{i=1}^{Ns_k(t_g)} Tr_{ki}(t_g)\mathbf{1}(as_k(t_g) = \text{S3}) \quad (17) \end{aligned}$$

with,

$$\begin{aligned} Nmu_j(t_g) &= Nm_j(t_g) \\ &\quad - \sum_{l=1}^{ns_k(t_g)} \mathbf{1}(as_k(t_g) = \text{S2}). \quad (18) \end{aligned}$$

As MBSs are the only devices that are connected to the power grid in the HUDN, the grid power consumption of the network during time period $[t_g,\ t_g + \Delta t_s]$ can be

$$\begin{aligned} E_k(t_g) = &\left(\sum_{j=1}^{\lfloor \phi_{MBS} \rfloor} \sum_{k=1}^{Nm_j(t_g)} \left(p_{motr} G_{mo} h_{kj} d_{kj}^{-\alpha_p}\right)\mathbf{1}(am_j(t_g) = \text{M2}) \right. \\ &\left. + \sum_{j=1}^{\lfloor \phi_{MBS} \rfloor} \left(p_{mhtr} G_{mh} h_{kj} d_{kj}^{-\alpha_p}\right)\mathbf{1}(am_j(t_g) = \text{M3}) \right)\mathbf{1}(as_k(t_g) = \text{S2})\Delta t_s \\ &+ \left(\sum_{n=1}^{\lfloor \phi_{TV} \rfloor} p_{tvtr} \cdot 10^{\frac{-Pl_{kn}}{10}} \right)\mathbf{1}(as_k(t_g) = \text{S1})\Delta t_s \end{aligned} \quad (12)$$

$$Tr_{ji,o}(t_g) = Ba_{mo}\log_2\left(1 + \frac{p_{motr} G_{mo} h_{ji} d_{ji}^{-\alpha_p}}{p_{no} + \sum_{z=1}^{\lfloor \phi_{MBS} \rfloor} \left(p_{motr} G_{mo} h_{zi} d_{zi}^{-\alpha_p}\right)\mathbf{1}(am_z(t_g) = \text{M2})}\right)\Delta t_s \quad (16)$$

TABLE II
ACTIONS PATTERNS OF ue_i

Pattern U1	ue_i has no communication request and no associated base station with ue_i
Pattern U2	ue_i is associated with an MBS: mb_j, and $am_j(t_g) = \text{M1}$ or $am_j(t_g) = \text{M3}$
Pattern U3	ue_i is associated with an MBS: mb_j, and $am_j(t_g) = \text{M2}$
Pattern U4	ue_i is associated with an SBS sb_k with a battery level higher than B_{tr}, and $sb_k(t_g) = \text{S3}$

represented as,

$$Ef_{hn}(t_g) = \frac{Tr_{hn}(t_g)}{\sum_{j=1}^{\lfloor \phi_{MBS} \rfloor} pm_j(t_g)\Delta t_s}. \tag{19}$$

III. REINFORCEMENT LEARNING FRAMEWORK

A. Energy Efficiency Optimization

We assume that the cloud agent can fully control the actions of the MBSs and SBSs. Thus, at the beginning of each time slot, the cloud agent will decide which action to choose for each MBS and SBS. By defining the action set of the cloud agent as $\mathbb{S}_{\mathrm{A}}$, and the action taken by the cloud agent at time slot: t_g, as $\mathbf{a}(t_g)$, the action sequence that can be taken by the cloud agent from time slot 0 to t_n can be defined as,

$$A(t_n) = \{\mathbf{a}(0), \mathbf{a}(\Delta t_s), \ldots, \mathbf{a}(t_g), \ldots, \mathbf{a}(t_n - \Delta t_s), \mathbf{a}(t_n) | \mathbf{a}(t_g) \in \mathbb{S}_{\mathrm{A}}\}. \tag{20}$$

with

$$\mathbf{a}(t_g) = \left[am_1(t_g), \ldots, am_{\lfloor \phi_{MBS} \rfloor}(t_g), as_1(t_g), \ldots, as_{\lfloor \phi_{SBS} \rfloor}(t_g)\right]^{\mathbf{T}}, \tag{21}$$

where $\mathbf{T}$ is the transpose operation.

Here, we define the location of each UE, MBS, and SBS at time slot: t_g, as a vector $slm(t_g)$, $sls(t_g)$, and $slu(t_g)$, respectively. They can be expressed as,

$$\begin{aligned} slm(t_g) &= \left[xm_1(t_g), \ldots, xm_{\lfloor \phi_{MBS} \rfloor}(t_g); ym_1(t_g), \ldots, ym_{\lfloor \phi_{MBS} \rfloor}(t_g)\right]^{\mathbf{T}}, \\ sls(t_g) &= \left[xs_1(t_g), \ldots, xs_{\lfloor \phi_{SBS} \rfloor}(t_g); ys_1(t_g), \ldots, ys_{\lfloor \phi_{SBS} \rfloor}(t_g)\right]^{\mathbf{T}}, \\ slu(t_g) &= \left[xu_1(t_g), \ldots, xu_{\lfloor \phi_{UE} \rfloor}(t_g); yu_1(t_g), \ldots, yu_{\lfloor \phi_{UE} \rfloor}(t_g)\right]^{\mathbf{T}}, \end{aligned} \tag{22}$$

where $xm_j(t_g)$ and $ym_j(t_g)$ are the coordinates of the location of mb_j at t_g. Similarly, $xs_k(t_g)$ and $ys_k(t_g)$ are the coordinates of the location of sb_k at t_g. By denoting the element at position line x row y of the matrix Mat as $\langle Mat \rangle_{x,y}$, we have $\langle slm(t_g) \rangle_{mx} \subset \mathbb{A}_T$, $\langle sls(t_g) \rangle_{sx} \subset \mathbb{A}_T$, and $\langle slu(t_g) \rangle_{ux} \subset \mathbb{A}_T$ for arbitrary $1 \le mx \le \lfloor \phi_{MBS} \rfloor$, $1 \le sx \le \lfloor \phi_{SBS} \rfloor$, and $1 \le ux \le \lfloor \phi_{UE} \rfloor$.

Similarly, by defining the battery level of each SBS at t_g as a vector: $\mathbf{ba}(t_g)$, then $\mathbf{ba}(t_g)$ can be shown as,

$$\mathbf{ba}(t_g) = \left[bl_1(t_g), \ldots, bl_{\lfloor \phi_{SBS} \rfloor}(t_g)\right]^{\mathbf{T}}, \tag{23}$$

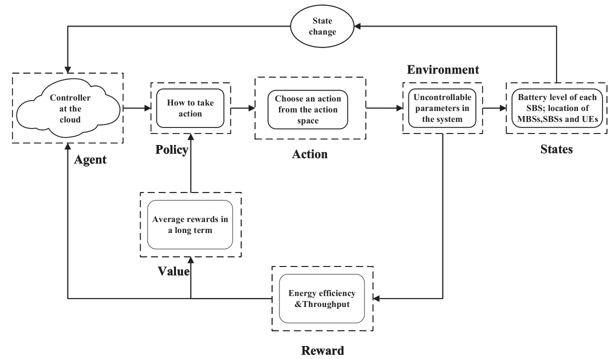

Fig. 2. Framework of the RL system.

and obviously, we have $0 \le \langle ba(t_g) \rangle_{bx} \le bl_{\max}$, for arbitrary $1 \le bx \le \lfloor \phi_{SBS} \rfloor$, where $bl_{\max}$ is the maximum capacity of the battery of an SBS.

In order to maximize the average energy efficiency of the network, the optimization problem can be formulated as,

$$\begin{aligned} Ef_{hn}^* = \arg\max_{A(t_n)} \ & \frac{\sum_{t_n=0}^{t_n} Ef_{hn}(t_g)}{t_n + 1} \\ \text{subject to} \ & \mathbf{a}(t_g) \in \mathbb{S}_{\mathrm{A}}, \\ & \langle slm(t_g) \rangle_m \subset \mathbb{A}_T, \quad \forall m \in [1, \lfloor \phi_{MBS} \rfloor], \\ & \langle sls(t_g) \rangle_s \subset \mathbb{A}_T, \quad \forall s \in [1, \lfloor \phi_{SBS} \rfloor], \\ & \langle slu(t_g) \rangle_u \subset \mathbb{A}_T, \quad \forall u \in [1, \lfloor \phi_{UE} \rfloor], \\ & \langle ba(t_g) \rangle_b \subset \mathbb{A}_T, \quad \forall b \in [0, bl_{\max}]. \end{aligned} \tag{24}$$

We should point out that since only MBSs are connected to the power grid, the service provider will be responsible for their energy consumption costs during the HUDN operation. So we only consider the energy consumption of the MBSs in (24).

B. Reinforcement Framework

The best action sequence to maximize the average energy efficiency depends on many uncontrollable, non-deterministic, and time-varying conditions. These include the location of UEs, SBSs, MBSs, and TV towers, as well as the mobility of the UEs, and whether they have communication requests, the battery level of each SBS at each time slot, and so on. The optimization problem in (24) is obviously NP-hard. On the other hand, as all conditions are memoryless (i.e., Markovian), the optimization problem can be considered as Markov decision process (MDP). Fortunately, as one of the most popular machine learning techniques, the reinforcement learning (RL) approach can be efficiently applied to MDP. So, we use RL to solve the optimization problem in (24).

In our approach we define the state set $\mathbb{S}_{\mathrm{S}}$ as formed by all possible battery level values of SBSs and all possible locations of MBSs, SBSs, and UEs. We can then see that $\mathbb{S}_{\mathrm{S}} \subset \mathbb{R}^3$, i.e., the state set is a subspace of three-dimensional real space because all values in the state set are continuous. Moreover, notice that there are $3^{\lfloor \phi_{MBS}+\phi_{SBS} \rfloor}$ types of actions for the agent to choose from at the beginning of each time slot, so the action set $\mathbb{S}_{\mathrm{A}}$ is very large. Therefore, we use a W-DDPG-based algorithm to perform optimization in (24). Notice that under these conditions, the action space is a large discrete space. However, the DDPG is used for actions with continuous values. So, in our approach we adopt the method in [36] to map the continuous action set to discrete action sets.

C. Brief Introduction of DDPG

DDPG is a reinforcement learning framework that can handle the continuous action sets based on the original actor-critic algorithm. As DDPG is the advanced algorithm of the actor-critic algorithm, it has four types of neural networks: 1) the online actor net, 2) the target actor net, 3) the online critic net, and 4) the target critic net. The architecture of the online actor net is the same as the target actor net. Also, the architecture of the online critic net is the same as the target critic net. Each of these four neural networks is constructed with several fully connected neural layers, and all layers contain their corresponding parameters. All parameters in a neural network are denoted as θ. The critic net is used to approximate the Q-table by using neural networks, while the actor net is trained to generate a deterministic policy, which is different from the stochastic policy gradient algorithm that chooses a random action from a giving distribution. Given the instantaneous state $\mathbf{s}(t_g) \in \mathbb{S}_{\mathrm{S}}$ and the action $\mathbf{a}(t_g) \in \mathbb{S}_{\mathrm{A}}$, if the policy of actor: μ, is deterministic, the Q value under policy μ can be expressed as

$$\begin{aligned} Q^{\mu}(\mathbf{s}_t, \mathbf{a}_t) = \mathbb{E}_{r_t, \mathbf{s}_{t+1} \sim \psi}[r(\mathbf{s}_t, \mathbf{a}_t) \\ + \gamma[Q^{\mu}(\mathbf{s}_{t+1}, \mu(\mathbf{s}_{t+1}))]] \end{aligned} \tag{25}$$

To simplify the above expressions, we use $\mathbf{s}_t$ to denote $\mathbf{s}(t_g)$ and $\mathbf{s}_{t+1}$ to represent $\mathbf{s}(t_g)$ and $\mathbf{s}(t_g + \Delta t_s)$, respectively. Similarly, $\mathbf{a}(t_g)$ and $\mathbf{a}(t_g + \Delta t_s)$ are replaced by $\mathbf{a}_t$ and $\mathbf{a}_{t+1}$, respectively. $r(\mathbf{s}_t, \mathbf{a}_t)$ is the reward of the state-action pair $(\mathbf{s}_t, \mathbf{a}_t)$, and r_t is the reward at time slot *t*. γ stands for the discount factor in Bellman equation, and ψ is the corresponding expectation distribution for $\mathbf{s}_{t+1}$ and r_t.

Based on the Bellman equation, the loss of the critic net is defined as

$$Lo\left(\theta^Q\right) = \mathbb{E}_{\mathbf{s}_t \sim \rho^{\psi}, \mathbf{a}_t \sim \psi, r_t \sim Ev}\left[\left(Q\left(\mathbf{s}_t,\ \mathbf{a}_t \middle| \theta^Q\right) - y_t\right)^2\right] \tag{26}$$

where ρ^{ψ} corresponds to the distribution of the state $\mathbf{s}_t$ under the current deterministic policy ψ, and *Ev* represents the environment. θ^Q is a parameter vector that includes the weights of all neurons in the online critic network. y_t in (26) is defined as follows,

$$y_t = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma Q\left(\mathbf{s}_{t+1},\ \mu(\mathbf{s}_{t+1}) \middle| \theta^Q\right). \tag{27}$$

The policy of the actor net will be updated based on the output of the critic net, where the gradient-based method is used to update the online actor net as (28), shown at the bottom of the page, where θ^{μ} is the parameter vector of the online actor net.

The training process can be described as follows.

First, with action $\mu(\mathbf{s}_t)$ given by the actor net, a noise n_t will be added to $\mu(\mathbf{s}_t)$ by the DDPG agent, and the action becomes $\mathbf{a}_t = \mu(\mathbf{s}_t) + n_t$. After action $\mathbf{a}_t$ is taken, the DDPG agent will observe a reward r_t and the next state $\mathbf{s}_{t+1}$ (changed from $\mathbf{s}_t$ due to the interaction between the agent and the environment *Ev*). Then, DDPG will store the experience set $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$ in the experience replay buffer **B**. Subsequently, N_{ba} sets of experiences are randomly selected by the DDPG agent from buffer **B** to construct a mini-batch. Simultaneously, N_{ba} sets of experiences are transferred into both the actor net and critic net. Subsequently, the actor target net outputs action $\mu'(\mathbf{s}_t + 1)$ based on $\theta^{\mu'}$ to the critic target net. According to the experience sets in the minibatch and $\mu'(\mathbf{s}_t + 1)$, the target critic net can calculate y_t based on (27) and input it to the online critic net [5].

With a given optimizer, e.g., Adam optimizer in this article, the online critic net will be updated. Afterwards, the online actor net gives action $\mu(\mathbf{s}_t)$ to the online critic net to achieve the gradient of the corresponding action, $\nabla_{\mathbf{a}} Q(\mathbf{s},\ \mathbf{a}|\theta^Q)|_{\mathbf{s}=\mathbf{s}_t, \mathbf{a}=\mu(\mathbf{s}_t)}$. With the optimizer of the actor net, the parameter gradient of θ^{μ} can be derived by $\nabla_{\theta^{\mu}} \mu(\mathbf{s}|\theta^{\mu})|_{\mathbf{s}=s_t}$. Based on the two gradients, i.e., $\nabla_{\mathbf{a}} Q(\mathbf{s},\ \mathbf{a}|\theta^Q)|_{\mathbf{s}=\mathbf{s}_t, \mathbf{a}=\mu(\mathbf{s}_t)}$ and $\nabla_{\theta^{\mu}} \mu(\mathbf{s}|\theta^{\mu})|_{\mathbf{s}=\mathbf{s}_t}$, the online actor net will be updated with the approximation as (29), shown at the bottom of the page [28].

Finally, DDPG updates the target nets in both the critic and actor net with a small constant τ, i.e.,

$$\begin{aligned} \theta^{Q'} &\leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'} \\ \theta^{\mu'} &\leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}. \end{aligned} \tag{30}$$

D. Wolpertinger Based DDPG

The Wolpertinger based DDPG (W-DDPG) is first proposed in [36] to assist deep reinforcement learning in large discrete action sets. The W-DDPG architecture used to map the output of a neural network from a continuous space $\mathbb{R}^n$ to a discrete

$$\nabla_{\theta^{\mu}} J \approx \mathbb{E}_{\mathbf{s}_t \sim \rho^{\psi}}\left[\nabla_{\mathbf{a}} Q\left(\mathbf{s},\ \mathbf{a} \middle| \theta^Q\right)\Big|_{\mathbf{s}=\mathbf{s}_t, \mathbf{a}=\mu(\mathbf{s}_t)} \nabla_{\theta^{\mu}} \mu(\mathbf{s}|\theta^{\mu})|_{\mathbf{s}=\mathbf{s}_t}\right] \tag{28}$$

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N_{ba}} \sum_{i=1}^{N_{ba}} \left[\nabla_{\mathbf{a}} Q\left(\mathbf{s},\ \mathbf{a} \middle| \theta^Q\right)\Big|_{\mathbf{s}=\mathbf{s}_i, \mathbf{a}=\mu(\mathbf{s}_i)} \nabla_{\theta^{\mu}} \mu(\mathbf{s}|\theta^{\mu})|_{\mathbf{s}=\mathbf{s}_i}\right] \tag{29}$$

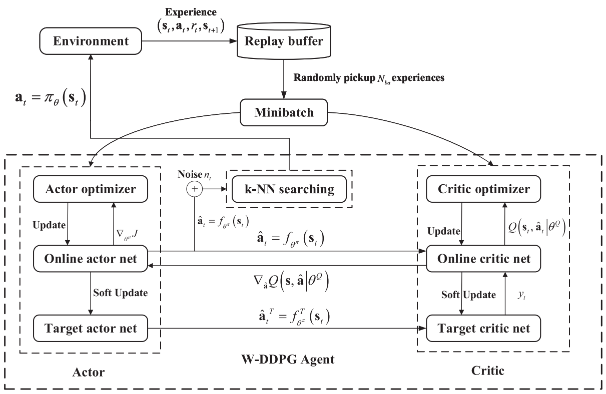

Fig. 3. W-DDPG architecture.

action space $\mathbb{S}_{\mathrm{A}}$. First, let's consider the proto-action, which is generated by the online actor net of a W-DDPG agent,

$$\begin{aligned} f_{\theta^\pi} &: S \rightarrow \mathbb{R}^n, \\ f_{\theta^\pi}(\mathbf{s}_t) &= \hat{\mathbf{a}}_t, \end{aligned} \tag{31}$$

where θ^π is the parameter of the online actor net of W-DDPG, $f_{\theta^\pi}(\cdot)$ is a function parametrized by θ^π, mapping from the state representation space to the action representation space: $\mathbb{R}^n$, and $\hat{\mathbf{a}}_t \in \mathbb{R}^n$ is the continuous proto-action at time slot *t*. This function provides a proto-action $\hat{\mathbf{a}}_t$ in $\mathbb{R}^n$ for a given state $\mathbf{s}_t$. However, in most cases, proto-action is not a valid action, i.e., we have $\hat{\mathbf{a}}_t \notin \mathbb{S}_{\mathrm{A}}$. Therefore, a new method is needed to map from $\hat{\mathbf{a}}_t$ to an element in $\mathbb{S}_{\mathrm{A}}$. In order to address this, we have,

$$\begin{aligned} g &: \mathbb{R}^n \rightarrow \mathbb{S}_{\mathrm{A}}, \\ g_k(\hat{\mathbf{a}}_t) &= \underset{\mathbf{a}_t \in \mathbb{S}_{\mathrm{A}}}{\overset{k}{\arg\min}} \, |\mathbf{a}_t - \hat{\mathbf{a}}_t|_2, \end{aligned} \tag{32}$$

where g_k is a *k*-nearest-neighbor (k-NN) mapping from a continuous space to a discrete set. It returns the *k* actions in $\mathbb{S}_{\mathrm{A}}$ that are closest to $\hat{\mathbf{a}}_t$ by L2 distance. We use *k*-dimensional tree searching to find the k-NN of $\hat{\mathbf{a}}_t$ [37], which has sub-linear temporal complexity.

However, sometimes actions with low Q-values may be the closest action to $\hat{\mathbf{a}}_t$ even in a part of the space where most actions have a high Q-value. Moreover, the selected low Q-value actions may misguide the agent with the real Q-value of $\hat{\mathbf{a}}_t$, and further reduce the performance of the whole algorithm. Thus, simply generating the actions based on (32) is not ideal. To avoid picking these low Q-value actions, we can improve the choice of action by selecting those with the highest Q-value based on $Q(\mathbf{s}_t, \mathbf{a}_t|\theta^Q)$,

$$\pi_\theta(\mathbf{s}_t) = \underset{\mathbf{a}_t \in g_k(f_{\theta^\pi}(\mathbf{s}_t))}{\arg\max} \; Q\Big(\mathbf{s}_t, \mathbf{a}_t \Big| \theta^Q\Big), \tag{33}$$

where π_θ is the policy of the W-DDPG. By defining the action generated by the target actor net as $\hat{\mathbf{a}}_t^T = f_{\theta^T}(\mathbf{s}_t)$, the architecture of the W-DDPG is shown in Fig. 3.

Here, we use an Ornstein-Uhlenbeck based process to generate exploration noise [28], in order to ensure the agent looks for other possible actions. The generating process of the noise term n_t is defined as $\mathcal{N}$. The state, reward, and action in the algorithm are defined as follows;

State: The state of the HUDN at time slot, *t*, (i.e., $\mathbf{s}_t$) is defined as a matrix that includes the information on the positions of MBSs, SBSs, and UEs, as well as the battery levels of SBSs. It can be expressed as,

$$\mathbf{s}_t = \begin{bmatrix} xm_1(t) & ym_1(t) & 0 \\ \vdots \lfloor \phi_{MBS} \rfloor & \vdots \lfloor \phi_{MBS} \rfloor & \vdots \lfloor \phi_{MBS} \rfloor \\ xs_1(t) & ys_1(t) & bl_1(t) \\ \vdots \lfloor \phi_{SBS} \rfloor & \vdots \lfloor \phi_{SBS} \rfloor & \vdots \lfloor \phi_{SBS} \rfloor \\ xu_1(t) & yu_1(t) & 0 \\ \vdots \lfloor \phi_{UE} \rfloor & \vdots \lfloor \phi_{UE} \rfloor & \vdots \lfloor \phi_{UE} \rfloor \\ xu_{\lfloor \phi_{UE} \rfloor}(t) & xu_{\lfloor \phi_{UE} \rfloor}(t) & 0 \end{bmatrix}. \tag{34}$$

Action: The action of the W-DDPG agent at time slot *t*, i.e., $\mathbf{a}_t$ is defined as a vector that includes information of the positions of MBSs, SBSs, and UEs, as well as the battery level of each SBS. It can be expressed as,

$$\begin{aligned} \mathbf{a}_t = \Big[& am_1(t),\ am_2(t), \ldots,\ am_{\lfloor \phi_{MBS} \rfloor}(t), \\ & as_1(t),\ as_2(t), \ldots,\ as_{\lfloor \phi_{SBS} \rfloor}(t) \Big]^{\mathbf{T}}, \end{aligned} \tag{35}$$

with

$$am_j(t) = \begin{cases} 1 & \text{if M1 action is selected} \\ 2 & \text{if M2 action is selected} \\ 3 & \text{if M3 action is selected} \end{cases}, \tag{36}$$

and

$$as_k(t) = \begin{cases} 1 & \text{if S1 action is selected} \\ 2 & \text{if S2 action is selected} \\ 3 & \text{if S3 action is selected} \end{cases}. \tag{37}$$

Reward: The reward received from the environment at time slot, *t* (i.e., r_t) is defined as $r_t = Ef_{hn}(t)$.

Therefore, the corresponding algorithm can be presented as Algorithm 1.

IV. Simulation Results

In this article, the actor net and critic net each has two hidden layers of fully-connected units with 500 and 400 neurons. The capacity of memory **B** is set at 10000 and the size of the mini batch is set as $N_{ba} = 128$. The discount factor γ is set as 0.95. The small constant τ is configured as 0.001, and the exploration rate ε is set as 0.01. Based on [31], [32], [35], the parameters of power consumption models (5) and (6) are set as $p_{str} = 1$ W, $\eta_{pa} = 8$, $p_{srf} = 0.7$ W, $p_{sbb} = 1.6$ W, $\sigma_{dc} = 0.08$, $\sigma_{ms} = 0.1$, $p_{sst} = 5$ W, $p_{mhtr} = 20$ W, $p_{motr} = 4$ W, $p_{mhrf} = 10$ W, $p_{morf} = 4$ W, $p_{mhbb} = 10$ W, $p_{mobb} = 4$ W and $p_{mst} = 1000$ W. The Wolpertinger architecture is set to search for the 100 nearest neighbors of the proto-action. Other defaulted parameters are configured as Table III.

The average energy efficiency, Ef_{hn}^*, with respect to the number of different RL methods, is shown in Fig. 4. We compare the proposed W-DDPG algorithm with other two algorithms, i.e., the regular DDPG and DQL. As the outputs

Algorithm 1 W-DDPG

Randomly initialize critic net Q_{θ^Q} and actor net f_{θ^π} with weights θ^Q and θ^π;
Initialize target networks $Q_{\theta^{Q'}}$ and f_{θ^T} with weights $\theta^{Q'} \leftarrow \theta^Q$ and $\theta^T \leftarrow \theta^\pi$;
Initialize replay buffer $\mathbf{B}$;
for $episode = [1,\ Max_e]$ **do**
  Initialize a random process $\mathcal{N}$ for action exploration;
  Receive initial observation state $\mathbf{s}_1$;
  for $t = [1,\ Max_t]$ **do**
    Select action $\mathbf{a}_t = \pi_\theta(\mathbf{s}_t)$ according to the current policy;
    Execute action $\mathbf{a}_t = \pi_\theta(\mathbf{s}_t)$ and observe reward r_t and new state $\mathbf{s}_{t+1}$;
    Store transition $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$ in $\mathbf{B}$;
    Sample a random minibatch of N_{ba} experiences $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$ from $\mathbf{B}$;
    Set $y_t = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma Q\Big(\mathbf{s}_{t+1},\ f_{\theta^T}(\mathbf{s}_{t+1})\Big|\theta^{Q'}\Big)$;
    Update the critic by minimizing the loss:

$$Lo\left(\theta^Q\right) = \frac{1}{N_{ba}} \sum_{i}^{N_{ba}} \left(y_i - Q\left(\mathbf{s}_i,\ \mathbf{a}_i \middle| \theta^Q\right)\right)^2;$$

    Update the actor using the sampled gradient:

$$\nabla_{\theta^\pi} J \approx \frac{1}{N_{ba}} \sum_{i=1}^{N_{ba}} \left[\nabla_{\mathbf{a}} Q\left(\mathbf{s},\ \hat{\mathbf{a}} \middle| \theta^Q\right)\Big|_{\mathbf{s}=\mathbf{s}_i, \hat{\mathbf{a}}=f_{\theta^\pi}(\mathbf{s}_i)} \cdot \nabla_{\theta^\pi} f_{\theta^\pi}(\mathbf{s})|_{\mathbf{s}=\mathbf{s}_i}\right];$$

    Update the target networks softly:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$
$$\theta^T \leftarrow \tau\theta^\pi + (1-\tau)\theta^T$$

  end for
end for

TABLE III
VALUES OF SYMBOLS USED IN SIMULATION

Symbol	Definition/explanation	Value
Max_e	Maximum training episodes	500
Max_t	Maximum training steps	1000
α_p	Path loss exponent	3.5
Δt_s	Length of a time slot	1 s
p_{tvtr}	Transmit power of TV towers	960 Kw
ft_n	Transmission frequency of tv_n	512-524 Mhz
Ht_n	Height of the TV tower tv_n	114-125 m
d_{kn}	Distance between tv_n and sb_k	1-3 Km
Hs_k	Height of the receiving antenna of sb_k	10 m
G_{mo}	Antenna gain for omnidirectional transmissions	10 dB
G_{mh}	Antenna gain for hybrid precoding transmissions	180 dB
σ_n	The standard deviation of Gaussian noise	0.01
Ba_{mh}	Transmission bandwidth of action M1 and M3	2 Ghz
Ba_{sh}	Transmission bandwidth of action S3	2 Ghz
Ba_{mo}	Transmission bandwidth of action M2	10 Mhz
$bl_{\max}$	Capacity of the battery of SBSs	10000 J
B_{tr}	Threshold of battery level	1000 J

of a regular DDPG agent correspond to continuous actions, we adopt a simple method to discretize them. For an arbitrary $\langle\mathbf{a}_t\rangle_i$, if $0 \le \langle\mathbf{a}_t\rangle_i \le 3$, then $\langle\mathbf{a}_t\rangle_i \leftarrow Ro(\langle\mathbf{a}_t\rangle_i)$, where $Ro(\cdot)$ is the round function. If $\langle\mathbf{a}_t\rangle_i \le 0$ or $\langle\mathbf{a}_t\rangle_i \ge 3$, then $\langle\mathbf{a}_t\rangle_i$

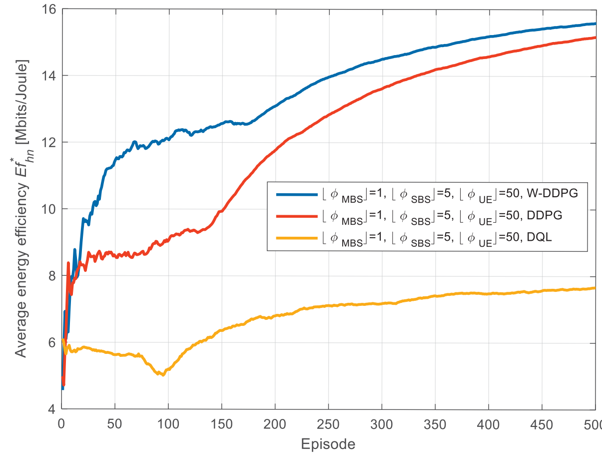

Fig. 4. Optimized average energy efficiency with respect to different RL methods.

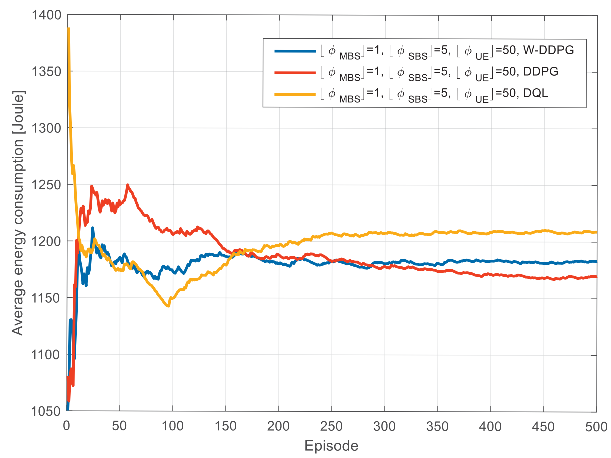

Fig. 5. Average energy consumption with respect to different RL methods.

will be clipped to ensure $0 \le \langle\mathbf{a}_t\rangle_i \le 3$. The structure of the neural net of the DQL algorithm used here is the same as in W-DDPG and DDPG, i.e., two hidden layers of fully-connected units with 500 and 400 neurons. As we can see, the energy efficiency of W-DDPG is the highest, while the energy efficiency of DQL is the lowest. This is mainly because the DQL algorithm is not suitable for solving tasks with large action spaces. On the other hand, compared with the original DDPG algorithm, the k-NN algorithm supported by W-DDPG can effectively prevent the agent outputting a low Q-valued action after the discretization process. Also, for all three algorithms, the average energy efficiency, Ef^*_{hn}, increases with a rise of the learning episode. This result indicates that our W-DDPG method can help the agent to achieve a better optimization on the energy efficiency of the HUDN.

An average energy consumption with respect to different RL methods is shown in Fig. 5. As can be observed, the gaps in average energy consumption among these three algorithms are not as large as the gaps in Ef^*_{hn}. This is because the value of p_{mst}, which is constant and cannot be optimized, is much larger than any other type of energy consumption. Thus, the influence of different algorithms on the average energy consumption is not significant when compared with the average energy efficiency Ef^*_{hn}.

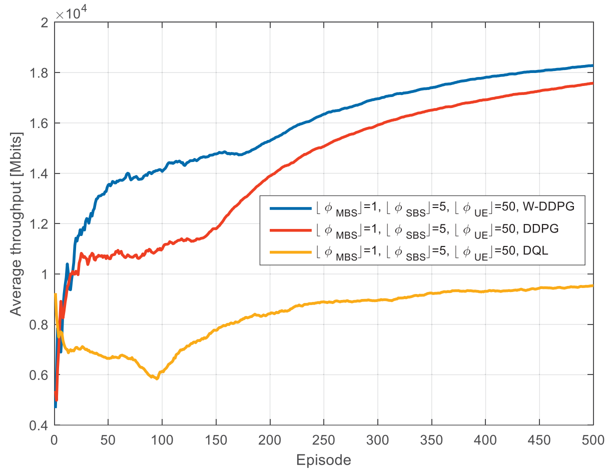

Fig. 6. Average throughput with respect to different RL methods.

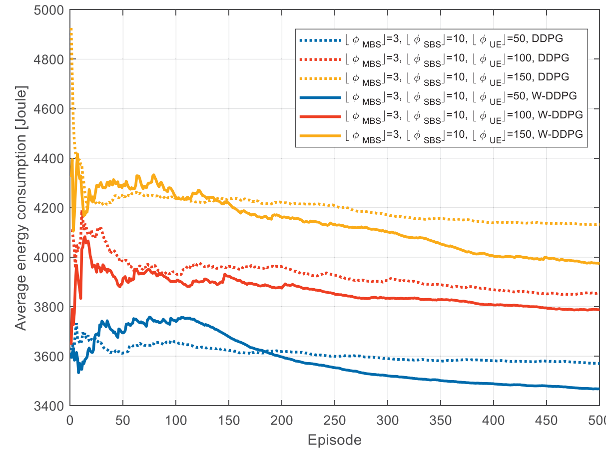

Fig. 8. Average energy consumption with respect to $\lfloor \phi_{UE} \rfloor$.

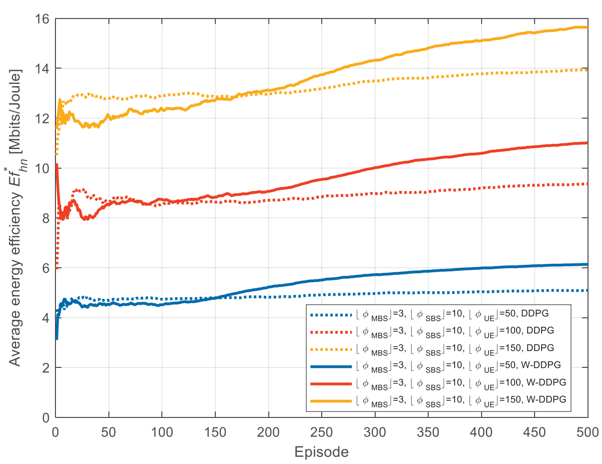

Fig. 7. Optimized average energy efficiency with respect to $\lfloor \phi_{UE} \rfloor$.

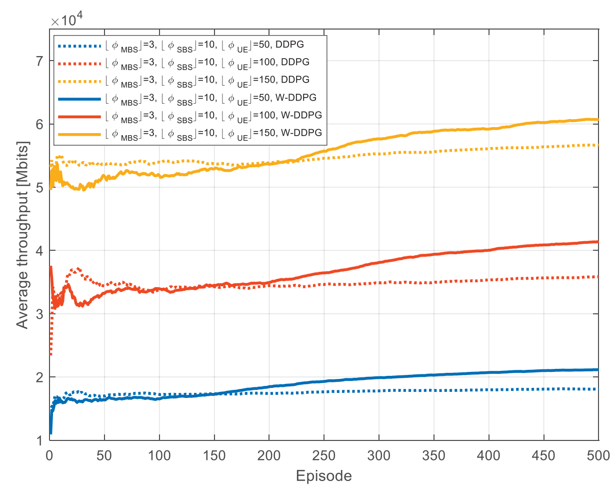

Fig. 9. Average throughput with respect to $\lfloor \phi_{UE} \rfloor$.

The average throughput with respect to different RL methods is shown in Fig. 6. The average throughput of the W-DDPG algorithm is the highest compared with the other two algorithms. Also, the simulation results in Fig. 6 seem to follow the same trend as in Fig. 4. This is also because of the large value of p_{mst}, which enables the agent to improve energy efficiency, hence reduce power consumption. As a result, the agent can optimize average energy efficiency, Ef^*_{hn}, by increasing the average throughput.

The average energy efficiency Ef^*_{hn} with respect to the number of UEs, $\lfloor \phi_{UE} \rfloor$ is depicted in Fig. 7, which shows how the energy efficiency increases as $\lfloor \phi_{UE} \rfloor$ increases. This is because the throughput of the networks mainly depends on the number of communication requests from UEs, while the energy consumption increases more slowly than the throughput. On the other hand, when $\lfloor \phi_{UE} \rfloor$ is fixed, we can see that Ef^*_{hn} increases at higher training episodes, which indicates the impact of the W-DDPG method. Also, similar to the result in Fig. 4, the performance of the W-DDPG algorithm is better than the performance of the DDPG algorithm. Moreover, as we can observe from Fig. 4 and Fig. 7, the performance gaps between these two algorithms increase with an increase in the size of the action space. This result verifies that it is more important to adopt W-DDPG for tasks with large action spaces.

The average energy consumption with respect to the number of UEs $\lfloor \phi_{UE} \rfloor$ is shown in Fig. 8. As shown, the energy consumption increases as $\lfloor \phi_{UE} \rfloor$ increases. On the other hand, when $\lfloor \phi_{UE} \rfloor$ is fixed, the energy consumption decreases at a higher training episode. Moreover, the decrease of average energy consumption is more noticeable when $\lfloor \phi_{UE} \rfloor = 150$. Since there are too many communication requests when $\lfloor \phi_{UE} \rfloor = 150$, all base stations in the network have to deal with heavy traffic loads. Under these conditions, energy consumption of the HUDN increases rapidly and this provides better opportunities for the RL agent to reduce energy consumption by selecting actions with higher average rewards. In contrast to the result in Fig. 5, gaps in average energy consumption between the W-DDPG algorithm and the DDPG algorithm become significant. This is mainly because any increase in the size of the action space makes the DDPG agent more likely to output actions with low Q-values.

Fig. 9 shows the average throughput with respect to the number of UEs $\lfloor \phi_{UE} \rfloor$. As can be seen, the average throughput increases with an increase of $\lfloor \phi_{UE} \rfloor$. However, when $\lfloor \phi_{UE} \rfloor$ is fixed, the throughput performance of the W-DDPG method

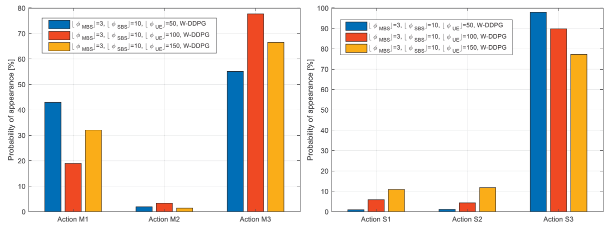

Fig. 10. Probability of action appearance with respect to $\lfloor \phi_{UE} \rfloor$.

becomes more significant when $\lfloor \phi_{UE} \rfloor = 100$. This is because the throughput of the network is restricted by an insufficient number of communication requests from UEs when $\lfloor \phi_{UE} \rfloor = 50$, while the number of base stations is not enough when $\lfloor \phi_{UE} \rfloor = 150$. Thus, the effect of the W-DDPG method is inadequate in both cases. On the other hand, when $\lfloor \phi_{UE} \rfloor = 100$, the number of UEs and base stations is more balanced. Therefore, in this case, the W-DDPG is capable of improving the performance. Also, similar to the results in Fig. 7 in terms of throughput the W-DDPG outperforms the DDPG algorithm.

Fig. 10 shows the probability of the appearance of actions with respect to $\lfloor \phi_{UE} \rfloor$. As we can observe, this is quite different with varying values of $\lfloor \phi_{UE} \rfloor$. This further verifies that the proposed RL-based structure can effectively change the policy to improve the performance of the network. Notice that as the value of $\lfloor \phi_{UE} \rfloor$ becomes larger, actions S1 and S2 become more likely to be selected by the RL agent. As the traffic demand and energy consumption of the HUDN rises with the increase of $\lfloor \phi_{UE} \rfloor$, the RL agent changes the policy to control the SBSs to harvest more energy. If the battery level of an SBS is below B_{tr}, the traffic will be handed off to the MBSs, which are more energy consuming. When $\lfloor \phi_{UE} \rfloor = 50$, the RL agent may prefer to allow more traffic to be carried by MBSs. Since the average distance between BSs and UEs is large, using MBSs with larger transmission power to carry more network traffic can effectively improve the throughput of the HUDN. When $\lfloor \phi_{UE} \rfloor = 100$, the average distance is small enough for SBSs to carry more network traffic. Thus, in this case the RL agent prefers to select more M3 actions to charge the SBSs and carry traffic simultaneously. When $\lfloor \phi_{UE} \rfloor = 150$, the number of UEs is too large for SBSs to handle because of battery limitation. Therefore, in this case more UEs will be automatically assigned to MBSs. Also, as more S1 actions are selected by SBSs, more energy is harvested from TV towers to reduce energy consumption from the power grid. Thus, less M3 actions are selected by MBSs, compared with the case of $\lfloor \phi_{UE} \rfloor = 100$.

The average battery level with respect to $\lfloor \phi_{UE} \rfloor$ is shown in Fig. 11. As we can see, when $\lfloor \phi_{UE} \rfloor$ is fixed, the average battery level converges with the rise of the training episode. Moreover, the converged battery level of the HUDN decreases as $\lfloor \phi_{UE} \rfloor$ increases. This is because an increase in traffic demand requires higher energy consumption. Another interesting result is that the average battery level with $\lfloor \phi_{UE} \rfloor = 150$ decreases at the beginning of training and then increases after the training episode becomes larger as more actions will be explored. More specifically, the agent concludes that energy efficiency is higher when more communication requests are carried by SBSs.

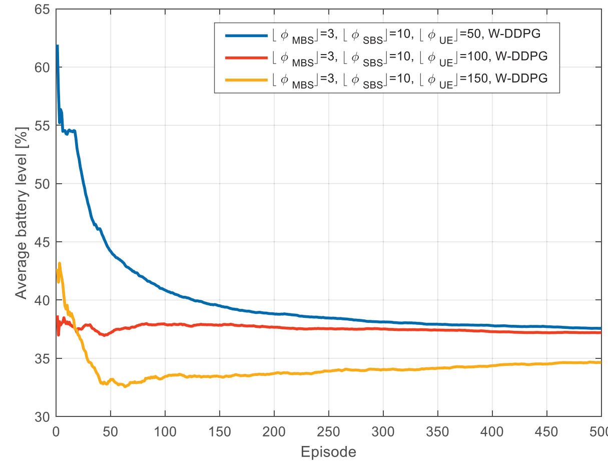

Fig. 11. Probability of action appearance with respect to $\lfloor \phi_{UE} \rfloor$.

V. Conclusion

This article mainly focuses on developing methods to optimize the energy charging efficiency of Heterogeneous ultra-dense networking (HUDN). Efficiently controlling data transmission and energy harvesting can profoundly influence the overall performance of HUDN. The main challenge is to how to optimally control both, which cannot be solved by regular optimization methods such as convex optimization. This is because the amount of harvested energy mainly depends on the transmission environment, which is highly random and difficult to predict. Advances in artificial intelligence (AI) technology can be utilized to solve the combined energy harvesting and communication optimization problem Therefore, in this article we first establish a theoretical network model to derive the optimization problem. Then, a reinforcement learning-based framework is considered to optimize the energy efficiency of the HetNet. We specifically develop a W-DDPG-based algorithm to deal with the large discrete action space in the learning task. The simulation results verify that the proposed W-DDPG-based method outperforms DQL, as well as the original DDPG based method.

References

[1] X. Ge, S. Tu, G. Mao, C. Wang, and T. Han, "5G ultra-dense cellular networks," *IEEE Wireless Commun.*, vol. 23, no. 1, pp. 72–79, Feb. 2016.

[2] S. Zhang, Q. Wu, S. Xu, and G. Y. Li, "Fundamental green tradeoffs: Progresses, challenges, and impacts on 5G networks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 33–56, 1st Quart., 2017.

[3] E. Hossain and M. Hasan, "5G cellular: Key enabling technologies and research challenges," *IEEE Instrum. Meas. Mag.*, vol. 18, no. 3, pp. 11–21, Jun. 2015.

[4] J. Huang, J. Cui, C.-C. Xing, and H. Gharavi, "Energy-efficient SWIPT-empowered D2D mode selection," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 3903–3915, Apr. 2020.

[5] C. Qiu, Y. Hu, Y. Chen, and B. Zeng, "Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8577–8588, Oct. 2019.

[6] X. Lu, P. Wang, D. Niyato, and E. Hossain, "Dynamic spectrum access in cognitive radio networks with RF energy harvesting," *IEEE Wireless Commun.*, vol. 21, no. 3, pp. 102–110, Jun. 2014.

[7] J. Fernandez-Bes, J. Cid-Sueiro, and A. G. Marques, "An MDP model for censoring in harvesting sensors: Optimal and approximated solutions," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 8, pp. 1717–1729, Aug. 2015.

[8] A. Minasian, S. ShahbazPanahi, and R. S. Adve, "Energy harvesting cooperative communication systems," *IEEE Trans. Wireless Commun.*, vol. 13, no. 11, pp. 6118–6131, Nov. 2014.

[9] M.-L. Ku, W. Li, Y. Chen, and K. J. R. Liu, "On energy harvesting gain and diversity analysis in cooperative communications," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2641–2657, Dec. 2015.

[10] T. Zhang, W. Chen, Z. Han, and Z. Cao, "A cross-layer perspective on energy-harvesting-aided green communications over fading channels," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1519–1534, Apr. 2015.

[11] F. Zhang and V. K. N. Lau, "Closed-form delay-optimal power control for energy harvesting wireless system with finite energy storage," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5706–5715, Nov. 2014.

[12] J.-M. Kang, C.-J. Chun, and I.-M. Kim, "Deep-learning-based channel estimation for wireless energy transfer," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2310–2313, Nov. 2018.

[13] D. He, C. Liu, H. Wang, and T. Q. S. Quek, "Learning-based wireless powered secure transmission," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 600–603, Apr. 2019.

[14] F. Azmat, Y. Chen, and N. Stocks, "Predictive modelling of RF energy for wireless powered communications," *IEEE Commun. Lett.*, vol. 20, no. 1, pp. 173–176, Jan. 2016.

[15] K. Wu, F. Li, C. Tellambura, and H. Jiang, "Optimal selective transmission policy for energy-harvesting wireless sensors via monotone neural networks," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9963–9978, Dec. 2019.

[16] J. Luo, J. Tang, D. K. C. So, G. Chen, K. Cumanan, and J. A. Chambers, "A deep learning-based approach to power minimization in multi-carrier NOMA with SWIPT," *IEEE Access*, vol. 7, pp. 17450–17460, 2019.

[17] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[18] C. W. Wang, Q. Xia, X. Yao, W. Wang, and J. M. Jornet, "Multi-hop deflection routing algorithm based on Q-learning for energy-harvesting nanonetworks," in *Proc. IEEE 15th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, 2018, pp. 362–370.

[19] R. C. Hsu and T. Lin, "A fuzzy Q-learning based power management for energy harvest wireless sensor node," in *Proc. Int. Conf. High Perform. Comput. Simulat. (HPCS)*, 2018, pp. 957–961.

[20] X. Zhang, J. Wang, and Q. Zhu, "Q-learning based energy harvesting for heterogeneous statistical QoS provisioning over multihop big-data relay networks," in *Proc. Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber Phys. Soc. Comput. (CPSCom) IEEE Smart Data (SmartData)*, Atlanta, GA, USA, 2019, pp. 807–814.

[21] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.

[22] Q. V. Do and I. Koo, "A transfer deep Q-learning framework for resource competition in virtual mobile networks with energy-harvesting base stations," *IEEE Syst. J.*, early access, Dec. 25, 2019, doi: 10.1109/JSYST.2019.2958993.

[23] H. Li, H. Gao, T. Lv, and Y. Lu, "Deep Q-learning based dynamic resource allocation for self-powered ultra-dense networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Kansas City, MO, USA, 2018, pp. 1–6.

[24] Y. Teng, M. Yan, D. Liu, Z. Han, and M. Song, "Distributed learning solution for uplink traffic control in energy harvesting massive machine-type communications," *IEEE Wireless Commun. Lett.*, vol. 9, no. 4, pp. 485–489, Apr. 2020.

[25] M. Li, X. Zhao, H. Liang, and F. Hu, "Deep reinforcement learning optimal transmission policy for communication systems with energy harvesting and adaptive MQAM," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5782–5793, Jun. 2019.

[26] M. Chu, H. Li, X. Liao, and S. Cui, "Reinforcement learning-based multiaccess control and battery prediction with energy harvesting in IoT systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2009–2020, Apr. 2019.

[27] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[28] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," 2015. [Online]. Available: arXiv:1509.02971.

[29] H. Ke, J. Wang, H. Wang, and Y. Ge, "Joint optimization of data offloading and resource allocation with renewable energy aware for IoT devices: A deep reinforcement learning approach," *IEEE Access*, vol. 7, pp. 179349–179363, 2019.

[30] H. Li, T. Lv, and X. Zhang, "Deep eterministic policy gradient based dynamic power control for self-powered ultra-dense networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, UAE, 2018, pp. 1–6.

[31] L. Wang, M. Elkashlan, R. W. Heath, M. Di Renzo, and K. Wong, "Millimeter wave power transfer and information transmission," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, 2015, pp. 1–6.

[32] I. R. Gomes, C. R. Gomes, H. S. Gomes, and G. P. D. S. Cavalcante, "Empirical radio propagation model for DTV applied to non-homogeneous paths and different climates using machine learning techniques," *PloS one*, vol. 13, no. 3, Mar. 2018, Art. no. e0194511.

[33] Y. Fang, I. Chlamtac, and Y.-B. Lin, "Channel occupancy times and handoff rate for mobile computing and PCS networks," *IEEE Trans. Comput.*, vol. 47, no. 6, pp. 679–692, Jun. 1998.

[34] M. Imran *et al.*, "Energy efficiency analysis of the reference systems, areas of improvements and target breakdown," ICT-EARTH deliverable, Rep., 2011.

[35] J. Ye, X. Ge, G. Mao, and Y. Zhong, "5G ultradense networks with nonuniform distributed users," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2660–2670, Mar. 2018.

[36] G. Dulac-Arnold *et al.*, "Deep reinforcement learning in large discrete action spaces," 2015. [Online]. Available: arXiv:1512.07679.

[37] C. Silpa-Anan and R. Hartley, "Optimised KD-trees for fast image descriptor matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, 2008, pp. 1–8.

Junliang Ye (Member, IEEE) received the B.Sc. degree in communication engineering from the China University of Geosciences, Wuhan, China, in 2011, and the Ph.D. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2018. He is currently serving as a Guest Researcher with the National Institute of Standards and Technology, U.S. Department of Commerce, Gaithersburg, MD, USA. His research interests include heterogeneous networks, stochastic geometry, mobility-based access models of cellular networks, millimeter-wave communications, and next-generation wireless communication.

Hamid Gharavi (Life Fellow, IEEE) received the Ph.D. degree from Loughborough University, U.K., in 1980. In 1982, he joined the Visual Communication Research Department, AT&T Bell Laboratories, Holmdel, NJ, USA. He was then transferred to Bell Communications Research (Bellcore) after the AT&T-Bell divestiture, where he became a Consultant on video technology and a Distinguished Member of Research Staff. In 1993, he joined Loughborough University as a Professor and the Chair of communication engineering. Since September 1998, he has been with the National Institute of Standards and Technology, U.S. Department of Commerce, Gaithersburg, MD, USA. He was a Core Member of Study Group XV (Specialist Group on Coding for Visual Telephony) of the International Communications Standardization Body CCITT and a member of the IEEE 2030 standard working group. His research interests include smart grid, wireless multimedia, mobile communications and wireless systems, mobile ad hoc networks, and visual communications. He received the Charles Babbage Premium Award from the Institute of Electronics and Radio Engineering in 1986, and the IEEE CAS Society Darlington Best Paper Award in 1989. He was a recipient of the Washington Academy of Science Distinguished Career in Science Award for 2017. He served as a Distinguished Lecturer of the IEEE Communication Society. He has been a Guest Editor for a number of special issues of the PROCEEDINGS OF THE IEEE, including Smart Grid, Sensor Networks & Applications, Wireless Multimedia Communications, Advanced Automobile Technologies, and Grid Resilience. He was the TPC Co-Chair for IEEE SmartGridComm in 2010 and 2012. He served as a member of the Editorial Board of PROCEEDINGS OF THE IEEE from January 2003 to December 2008. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and IEEE WIRELESS COMMUNICATIONS.